08/10/00 JC883 U.S. PTO

08-14-00

Microfiche

PTO/SB/05 (2/98) (modified)
Approved for use through 9/30/2000, OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

08/10/00 JC869 U.S. PTO 09/636299

NEW UTILITY PATENT APPLICATION TRANSMITTAL

(only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket Number	5111
First Named Inventor	Moshe B. Rubin
Total Pages in this Submission	70
Express Mail Label No.	EL482472075US

APPLICATION ELEMENTS	ACCOMPANYING APPLICATION PARTS
1. ☒ Fee Transmittal Form (in duplicate) ☒ Check Enclosed $424.00	6. ☒ Assignment & Assignment Recordation Cover Sheet
2. ☒ Specification (45 pages) *(preferred arrangement set forth below)* ■ Descriptive Title of the Invention ■ Cross Reference(s) to Related Case(s) ■ Statement Regarding Fed sponsored R & D ■ Background of the Invention ■ Brief Summary of the Invention ■ Brief Description of the Drawing(s) ■ Detailed Description ■ Claim or Claims ■ Abstract of the Disclosure	7. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)* 8. ☐ Information Disclosure Statement & PTO-1449 ☐ Copies of IDS Citation(s) 9. ☒ Preliminary Amendment 10. Small Entity Statement ☒ New Statement enclosed ☐ Statement filed in prior application. Status still proper and desired
3. ☒ Drawing(s) *(when necessary per 35 USC 113)* *(11 pages)*	11. ☒ Return Postcard
4. Oath or Declaration a. ☒ New Declaration ☒ Executed b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 17 completed)* i. ☐ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).	12. ☒ Microfiche 13. ☐ _____ 14. ☐ _____ 15. ☐ _____ 16. ☐ _____
5. ☐ Incorporation by Reference *(useable if Box 4b is checked)*. The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.	*ADDRESS TO:* **Box Patent Application** **Commissioner for Patents** **Washington, D.C. 20231**

17. **If a CONTINUING APPLICATION,** *check appropriate box and supply the requisite information below and in a preliminary amendment:*
☐ Continuation ☐ Divisional ☒ Continuation-in-part (CIP) of prior application No.: 09/459,493 (filed 12/13/99)
Prior application information: *Examiner:* Bryce P. Bonzo *Group/Art Unit:* 2785

18. CORRESPONDENCE ADDRESS

NAME	Laura A. Majerus Fenwick & West LLP				
ADDRESS	Two Palo Alto Square				
CITY	Palo Alto	*STATE*	CA	*ZIP CODE*	94306
COUNTRY	U.S.A.	*TELEPHONE*	(650) 858-7152	*FAX*	(650) 494-1417

Name (Print/Type)	Laura A. Majerus	*Registration No. (Attorney/Agent)*	33,417
Signature	Laura Majerus	*Date*	August 10, 2000

PTO/SB/ 10 (6-95) (modified)
Approved for use through 07/31/96. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN	Docket Number (Optional): 5111

Applicant or Patentee: Moishe Halibard and Moshe Binyamin Rubin

Application or Patent No.:

Filing Date or Issue Date:

Title: METHOD AND SYSTEM FOR COPY PROTECTION OF IMAGES DISPLAYED ON A COMPUTER MONITOR

I hereby declare that I am

[ ] the owner of the small business concern identified below:

[x] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Alchemedia Ltd.

ADDRESS OF SMALL BUSINESS CONCERN P.O. Box 400 Azor Tasia Har Tuv, Mercaz Ganir, Beit Shemesh 99100, Israel

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[ ] the specification filed herewith with title as listed above.

[X] the application identified above.

[ ] the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

[X] No such person, concern, or organization exists.

[ ] Each such person, concern or organization is listed below:

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING Daniel A. Schreiber

TITLE OF PERSON IF OTHER THAN OWNER Chief Executive Officer

ADDRESS OF PERSON SIGNING P.O. Box 400 Azor Tasia Har Tuv, Mercaz Ganir, Beit Shemesh 99100, Israel

SIGNATURE _____ DATE 30.7.00

IN THE UNITED STATES

PATENT AND TRADEMARK OFFICE

APPLICANT(S):	Rubin and Halibard
SERIAL NO.:	To Be Assigned
FILING DATE:	Herewith
TITLE:	Method and System for Copy Protection of Images Displayed on a Computer Monitor
EXAMINER:	To Be Assigned
GROUP ART UNIT:	To Be Assigned
ATTY. DKT. NO.:	21939-05111

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner For Patents, Washington, D.C. 20231, on the date shown below:

Dated: _____ By: _____
Laura A. Majerus, Reg. No.: 33,417

COMMISSIONER FOR PATENTS
WASHINGTON, DC. 20231

PRELIMINARY AMENDMENT

SIR:

Prior to the examination of the above-referenced application, please amend the application as described below:

Cancel Page 1, lines 4-7 and substitute therefor: --This application is a continuation-in-part of assignee's pending application U.S. Serial No. 09/459,493, filed December 13, 1999, entitled "Method and System for Copyright Protection of Digital Images."--.

Respectfully submitted,
RUBIN AND HALIBARD

Dated: August 10, 2000

By: Laura Majerus

Laura A. Majerus, Reg. No.: 33,417
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
Tel.: (650) 858-7152
Fax.: (650) 494-1417

DOCKET NO. 21939-05111

Method and System for Copy Protection of Images Displayed on a Computer Monitor

This application is a continuation-in-part of assignee's pending application U.S. Serial No. 09/397,331, filed on September 14, 1999, entitled "Method and System for Copyright Protection of Digital Images Transmitted over Networks."

FIELD OF THE INVENTION

The present invention relates to copy protection of digital data, and more specifically to copy protection of proprietary digital images displayed on a computer monitor.

BACKGROUND OF THE INVENTION

Software copy protection is a central concern in software development, and in copyright law itself. Typically, software is distributed in shrink-wrap packages containing diskettes and/or CD-ROMs, and by download over the Internet via ftp servers. Protecting software from rampant unauthorized copying, distribution and use ("software piracy") is one of the most challenging problems facing the software industry.

Over the past years, several techniques have been developed for combating software piracy. These include use of hardware plugs, use of license keys, use of tokens and sophisticated encryption systems.

One of the leading technologies for controlling use of software within turnkey transaction systems is the Digital Rights Management system of InterTrust® Technologies Corp. of Sunnyvale, CA, as described in U.S. Patents Nos. 5,892,900, 5,410,598, 5,050,213, 4,977,594 and 4,827,508. Information about InterTrust is available on the web at http://www.intertrust.com.

Another leading technology is the CyberSales Solution™ of SoftLock.com, Inc. of Maynard, MA, as described in U.S. Patent No. 5,509,070. CyberSales Solution provides locking and unlocking functionality so that content can be securely previewed by consumers, electronically purchased and redistributed, and it protects the content in an initial transaction and in subsequent information pass-along. Content providers can control how much information is available without paying, and disable, or additionally charge for, the ability to print or cut and paste. CyberSales Solution handles secure transactions, remittance processing, reports, audits and customer service. Information about CyberSales Solution is available on the web at http://www.softlock.com.

With the advent of the use of compelling multi-media on web pages accessible over the Internet, protection of digital images and other media is becoming increasingly critical. Web designers are reluctant to use valuable digital "works of art" knowing that users can easily copy them onto their own computers, and use them for their own unauthorized purposes. Moreover, anyone using a web browser to view an image posted on the Internet can easily copy the image by simply positioning a mouse pointer over the displayed image, clicking on the right mouse button and selecting a "Save Image As ..." command. Copyright and piracy issues are of major concern to web publishers.

Prior art techniques for protecting digital images include the embedding of invisible digital watermarks within images, so that copies of protected images can be identified and traced. Digimarc Corporation of Lake Oswego, OR embeds hidden messages within pixel data for identifying protected images, and tracks their distribution over the Internet to monitor potential copyright infringement. Digimarc images carry unique IDs that link to pre-determined locations on the web. Digimarc images are compatible with standard image formats, such as JPEG, and can be opened and displayed by standard image readers. However, when opened with a Digimarc reader, the images are displayed together with a "Web look up" button that enables a user to identify the sources of the images. Digimarc technology is described in U.S. Patents Nos. 5,862,260, 5,850,481, 5,841,978, 5,841,886, 5,832,119, 5,822,436, 5,809,160, 5,768,426, 5,765,152, 5,748,783, 5,748,763, 5,745,604, 5,721,788, 5,710,834 and 5,636,292. Information about Digimarc is available on the web at http://www.digimarc.com.

These techniques are useful in thwarting digital image piracy to the extent that they trace pirated content, but they do not prevent unauthorized copying of digital images in the first place.

Other prior art techniques require a webmaster to modify images residing on a server computer in order to protect them. The webmaster is also required to modify his web pages accordingly, so as to reference the modified images. SafeMedia™ is a software product of Internet Expression, Inc. of Exton, PA that converts images from a standard format such as JPEG into a SIF (Safe Image Format). SIF images can only be viewed with a SafeMedia Java viewer. SafeMedia embeds a host or domain name into an image, and checks that the image is located on the web site it was intended for. SafeMedia also includes enhanced system control for preventing screen capture by disabling a clipboard. Information about SafeMedia is available on the web at http://www.safemedia.com.

These prior art techniques are difficult to embrace, since they require modification of all protected images on the web, as well as modification of the web pages that reference them. Furthermore the SIF Java viewer has the

limitation of only being able to load images from the same server that the viewer came from.

Other prior art techniques for protecting digital images use Java applets within web browsers to disable the menu that pops up when a user right clicks on a displayed image within his web browser. Copysight® is a software application of Intellectual Protocols, LLC of Nanuet, NY that uses digital watermarking and fingerprinting to protect images, and includes a Java applet that disables the command to save displayed images within a web browser and the command to print them. Copysight operates by converting unprotected files to protected files that are encrypted and that contain digital fingerprints. Copysight also tracks distribution of protected images across the Internet, and issues reports of potential copyright infringement. Information about Copysight is available on the web at http://www.ip2.com.

Typically, when a digital image is displayed on a computer monitor, the pixel data is temporarily stored within a video RAM, the contents of which are then rendered on the monitor. Prior art techniques disable unauthorized copying of digital images from within web browsers, but they do not protect the images from being copied directly from a video RAM. For example, they do not prevent a user from copying digital images displayed in his web browser by means of a Print Screen or other such command that serves to capture contents of a video RAM to a clipboard. Thus a Java applet that prevents unauthorized copying of digital images from within Netscape Communicator or Internet Explorer can be circumvented by a user pressing on a Print Screen button of his keyboard, or by a user copying and pasting from a window of his web browser to a window of another software application.

Another way for a user to copy data from a video RAM within the Microsoft Windows® operating system is by use of a "transparent window" which does not erase its background. A transparent window is one that does not have its own background painted in it. Such a window, when opened on a screen, shows the contents of the screen transparently within it. That is, only its window frame appears on the screen, and the contents of the display appear through the window area within the frame. Although such a transparent window appears genuinely transparent, in fact the contents of the screen are copied and used as the background for the window. This is apparent when a user moves such a transparent window to a new location on the screen. When the window is moved, the previous portion of the screen appears in the window's new location.

Thus it can be appreciated that a user can copy an image displayed on a computer monitor by simply opening a transparent window and positioning it so that its frame encompasses the desired image. The contents of the display screen

within the frame are copied into the background of the window, making it possible for a user to then capture the contents, including the desired image.

Specifically, reference is made to Figure 1, which is a prior art illustration of a proprietary digital image 100. Typically, when image 100 is posted on a web page, such as HTML page 150 (Figure 2) residing on a server computer, and displayed by a web browser running on a client computer, a user can copy image 100 to his local computer.

Applicant's four co-pending applications

(i) U.S. Serial No. 09/313,067, filed on May 17, 1999 and entitled METHODS AND APPARATUS FOR PREVENTING REUSE OF TEXT, IMAGES AND SOFTWARE TRANSMITTED VIA NETWORKS;

(ii) U.S. Serial No. 09/397,331, filed on September 14, 1999 and entitled METHOD AND SYSTEM FOR COPYRIGHT PROTECTION OF DIGITAL IMAGES TRANSMITTED OVER NETWORKS;

(iii) U.S. Serial No. 09/459,493, filed on December 13, 1999 and entitled METHOD AND SYSTEM FOR COPYRIGHT PROTECTION OF DIGITAL IMAGES TRANSMITTED OVER NETWORKS; and

(iv) U.S. Serial No. 09/493,023, filed on January 27, 2000 and entitled METHOD AND SYSTEM FOR CONTENT COMMERCIALIZATION

concern technology for preventing unauthorized copying of images.

Reference is now made to Figure 2, which is an illustration of the results of a screen capture when proprietary digital image 100 is being displayed on the screen, and when copy protection technology from the above referenced patent applications is deployed. When a screen capture application tries to copy pixel data of proprietary digital image 100, watermarked image data is substituted for proprietary digital image 100. Using the technology described in the above-referenced patent applications, a web administrator can ensure that when a user tries to copy image 100 from within web page 150, another image, such as watermarked image 200, is substituted for image 100 prior to capture of image pixel data. Thus, although image 100 is displayed on a computer screen, a screen capture program will only be able to capture pixels of image 200.

Reference is now made to Figure 3, which is a prior art illustration of a transparent window 300 with a non-erase background. Although a typical Windows application immediately erases its main window, this is not necessary, and it is possible to write an application that does not erase its background. At first glance, window 300 appears to be transparent, with no background. However, this is not the case.

Reference is now made to Figure 4, which is a prior art illustration of transparent window 300 when it is moved to another location, revealing that it had copied a part of the screen display into its background when it was opened in its

previous position. Indeed, as can be seen in Figure 4, window 300 is not really transparent, but rather has a background that coincides with the pixel data from the position where it was originally opened.

Reference is now made to Figures 5A – 5D, which illustrate steps for creating an application having a window with a non-erase background, such as transparent window 300 (Figure 3), using a Microsoft Visual C++ compiler wizard. Figures 5A – 5D are representative images of user interface screens from Microsoft Visual C++ version 6. The first step is to create a new Microsoft Foundation Class (MFC) AppWizard (exe) project, as illustrated by element 510 in window 520 entitled "New" in Figure 5A, used for creating new projects. After selecting such an MFC project, the user clicks "OK", then successively clicks on "Next>" and finally on "Finish", through to the end of the program. The next step is to add a new WM_ERASEBKGND Windows message handler to the View, as illustrated by element 530 in window 540 entitled "New Windows Message and Event Handlers for class CNoeraseView" in Figure 5B. After adding such a message handler to the View, the user clicks on "Add". The next step is to click on "Edit", in order to modify the message handler code. Specifically, the user changes message handler code from that illustrated in Figure 5C to that illustrated in Figure 5D. The last step is to compile the application. When the application is run, it will generate a transparent window such as transparent window 300.

The complete software listing for the application generated by the above steps is provided in Appendix A.

Such a transparent window as transparent window 300 can be used to capture proprietary image data from a computer screen. Reference is now made to Figure 6, which is a prior art illustration of use of a transparent window to copy proprietary digital image 100 from a web page 150. Transparent window 600 is sized and positioned so that it includes proprietary digital image 100 within its frame when it is opened. Window 600 appears to be transparent, and digital image 100 shows through its frame. However, the pixel data that shows through window 600 is actually copied into its background.

Reference is now made to Figure 7, which is a prior art illustration of a screen capture of a screen in which transparent window 600 has been moved to another location, when copy protection technology from the above referenced patent applications is deployed. As can be seen from Figure 7, window 600 has copied proprietary digital image 100 into its background. As can be further seen from Figure 7, the proprietary digital image was recognized as being within web page 150 and watermarked data was substituted therefor. However, it was not recognized as being within transparent window 600, and watermarked data was not substituted therefor. The screen capture succeeded in capturing proprietary image 100 within transparent window 600.

Reference is now made to Figure 8, which is a prior art illustration of a software paint program 800 that has copied the clipboard from Figure 7 into its own window, thereby capturing proprietary digital image 100. At this stage, the user has unrestricted access to proprietary digital image 100.

Thus it can be seen that use of a transparent window with a non-erase background works around security software that normally protects proprietary image data that is displayed from being captured, and, as such, it poses a threat to copy protection of image data over the Internet.

SUMMARY OF THE INVENTION

The present invention provides a method and system for preventing unauthorized copying of proprietary digital image data.

There is thus provided in accordance with a preferred embodiment of the present invention a method for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, including providing screen pixel data for rendering on a computer monitor, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin, detecting that a second window is going to be displayed on the computer monitor, determining the position and size of the second window, determining, based on the position and size of the second window, a portion of the screen pixel data wherein the first window is going to be covered by the second window, and replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

There is further provided in accordance with a preferred embodiment of the present invention a system for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, including a computer monitor on which screen pixel data is rendered, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin, an event detector detecting that a second window is going to be displayed on the computer monitor, a window processor for determining the position and size of the second window, and for determining, based on the position and size of the second window, a portion of the screen pixel data wherein the first window is going to be covered by the second window, and a pixel processor for replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

There is yet further provided in accordance with a preferred embodiment of the present invention a method for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, including providing screen pixel data for rendering on a computer monitor, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin, detecting that a second window is going to be displayed on the computer monitor, determining the position and size of the second window, determining, based on the position and size of the second window, a portion of the screen pixel data wherein the proprietary digital image data is going to be covered by the second window, and replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

There is additionally provided in accordance with a preferred embodiment of the present invention a system for preventing copying of proprietary

digital image data that is rendered within a window displayed on a computer monitor, including a computer monitor on which screen pixel data is rendered, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin, an event detector detecting that a second window is going to be displayed on the computer monitor, a window processor for determining the position and size of the second window, and for determining, based on the position and size of the second window, a portion of the screen pixel data wherein the proprietary digital image data is going to be covered by the second window, and a pixel processor for replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

09653569.090100

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be more fully understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Figure 1 is a prior art illustration of a proprietary digital image;

Figure 2 is a prior art illustration of the results of a screen capture when the proprietary digital image of Figure 1 is being displayed on the screen;

Figure 3 is a prior art illustration of a transparent window with a non-erase background;

Figure 4 is a prior art illustration of the transparent window of Figure 3 when it is moved to another location, showing that it had copied a part of the screen display into its background;

Figures 5A – 5D are prior art illustrations of steps for creating an application having a window with a non-erase background, using a Microsoft Visual C++ compiler wizard;

Figure 6 is a prior art illustration of use of a transparent window to copy the proprietary digital image of Figure 1 from a web browser;

Figure 7 is a prior art illustration of a screen capture of a screen in which the transparent window of Figure 6 has been moved to another location;

Figure 8 is a prior art illustration of a software paint program that has copied the clipboard from Figure 7 into its own window, thereby capturing the proprietary digital image of Figure 1;

Figure 9 is a simplified illustration of a computer system for implementing a preferred embodiment of the present invention; and

Figure 10 is an illustration of the appearance of a window with a non-erase background encompassing a portion of a proprietary digital image.

LIST OF APPENDICES

Appendix A is a software listing of prior art functions for creating an application having a window with a non-erase background, using a Microsoft Visual C++ compiler.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

The present invention provides a method and system for preventing unauthorized copying of proprietary digital image data. Specifically, the present invention enables blocking use of windows with non-erase backgrounds for copying proprietary image data that is rendered on a computer monitor display screen.

As described hereinabove with reference to Figures 1 – 8, a window with a non-erase background can be used in order to circumvent prior art copy protection systems, by capturing proprietary digital image data within a transparent area of such a window. Although prior art copy protection systems can protect against various methods of screen capture of pixel data rendered on a display screen, such as screen capture by use of a PrintScreen command, or screen capture by copying screen data into a clipboard, the prior art systems do not protect against screen capture through use of a window with a non-erase background. The present invention overcomes this limitation.

Reference is now made to Figure 9, which is a simplified illustration of a computer system 900 for implementing a preferred embodiment of the present invention. As shown in Figure 9, computer system 900 includes an operating system 910, such as the Microsoft Windows operating system, which manages a plurality of software applications running on computer system 900. Proprietary pixels 920 from proprietary digital image 930 are rendered on a display screen 940 of a computer monitor 950, within an application window 960 such as a window of a web browser. Typically, pixel data rendered on display screen 940 is stored in a frame buffer 970 of a video card 980, and video card 980 rapidly generates pixel data for display screen 940 from the data in frame buffer 970. Typically the pixel data in frame buffer 970 is refreshed at rates of 30 frames per second or higher, and video card 970 accordingly refreshes display screen 940 at such rates. To protect proprietary image data, computer system 900 runs a copy protection application 990.

Reference is now made to Figure 10, which illustrates the appearance of a window 1020 with a non-erase background encompassing a portion of proprietary pixels 920 on display screen 940, such as window 600 that encompasses proprietary digital image 100 (Figure 6). Window 1020 can appear, for example, as a window of a newly running application program, or as a window of a running application that was previously not visible on display screen or previously positioned elsewhere. Window 1020 can appear in response to a user's launching of a software application, or resizing of a window, or maximizing of a window that was previously minimized, or bringing a window that was previously obscured behind window 960 in front of window 960.

In a preferred embodiment of the present invention, a software application, such as copy protection application 990, can use Windows system-wide

hooks to detect the appearance of window 1020 prior to its being rendered on display screen 940. Hooks are well known to those skilled in the art as a way to get information about operating system events. Specifically, with reference to the Windows operating system, copy protection application 990 preferably registers to have a Windows computer-based training (CBT) system-wide hook and a CallWndProc system-wide hook, which ensures that whenever operating system 910 receives a request for a new CBT event or a new WndProc event, copy protection application 990 is first notified prior to such event's occurrence. CBT events and WndProc events include opening of new windows, and appearances of windows on display screen 940. When operating system 910 receives a request to open window 1020, copy protection application 990 is first notified of such request, and intervenes prior to such request being effectuated by the system. With reference to Figure 10, copy protection application 990 is notified about window 1020 prior to window 1020 being opened in display screen 940.

In a preferred embodiment of the present invention, when copy protection application 990 receives notice of an event indicating that operating system 910 has been instructed to open window 1020, it determines an area of overlap between application window 960 and window 1020, based on the size and location of window 1020; namely, the hatched area of overlap 1030. Copy protection application 990 then replaces the pixel data in area of overlap 1030 with substitute pixel data, such as all white pixels, prior to the opening of window 1020. As such, when window 1020 does open, its background will only absorb the substitute pixel data in overlap area 1030, and not the proprietary pixel data. Thus proprietary pixels 920 remain copy protected.

If window 1020 is subsequently moved, its background remains the same, as illustrated in Figure 4, and thus its background continues to contain substitute pixel data instead of proprietary pixel data. However, there are several cases when window 1020 repaints its background, including:

(i) If window 1020 is minimized and subsequently opened, it repaints its background;

(ii) If window 1020 is enlarged, its background absorbs additional pixel data from the screen;

(iii) If window 1020 is reduced and subsequently enlarged, its background absorbs additional pixel data from the screen; and

(iv) If another application is made active so that its window partially or entirely covers window 1020, and window 1020 is subsequently brought in front of the other application's window, it repaints its background.

For each of these events, copy protection application 990 must be notified prior to the window event occurring, and it must replace pixel data in the appropriate overlap

area with substitute pixel data. Otherwise, window 1020 will absorb proprietary pixels 920 into its background.

It is not necessary for copy protection application 990 to replace pixel data with substitute pixel data unless window 1020 overlaps with application window 960. Moreover, it is not necessary to replace pixel data with substitute pixel data in the above cases (ii) and (iii) unless the additional portion of the background area of window 1020 after enlargement overlaps with application window 960. Similarly, it is not necessary to replace pixel data with substitute pixel data in the above case (iv), unless the portion of the background area of window 1020 that was covered by the other application window overlaps with application window 960.

It is appreciated that an alternative embodiment of the present invention is to modify the area of overlap between window 1020 and proprietary pixels 920, namely, the doubly hatched area of overlap 1040, rather than the (larger) area of overlap 1030 between window 1020 and window 950. In this alternative embodiment it is not necessary to replace pixel data with substitute pixel data unless window 1020 overlaps with proprietary pixels 920. Moreover, it is not necessary to replace pixel data with substitute pixel data in the above cases (ii) and (iii) unless the additional portion of the background area of window 1020 after enlargement overlaps with proprietary pixels 920. Similarly, it is not necessary to replace pixel data with substitute pixel data in the above case (iv), unless the portion of the background area of window 1020 that was covered by the other application window overlaps with proprietary pixels 920.

APPENDIX A

```
// MainFrm.h : interface of the CMainFrame class
//
/////////////////////////////////////////////////////////////////////////////

#if
!defined(AFX_MAINFRM_H__755A532D_2BDA_11D4_8C84_000086319872__I
NCLUDED_)
#define
AFX_MAINFRM_H__755A532D_2BDA_11D4_8C84_000086319872__INCLUD
ED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CMainFrame : public CMDIFrameWnd
{
DECLARE_DYNAMIC(CMainFrame)
public:
CMainFrame();

// Attributes
public:

// Operations
public:

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CMainFrame)
virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
//}}AFX_VIRTUAL

// Implementation
public:
virtual ~CMainFrame();
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif

protected:  // control bar embedded members
CStatusBar  m_wndStatusBar;
CToolBar    m_wndToolBar;

// Generated message map functions
protected:
//{{AFX_MSG(CMainFrame)
afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
   // NOTE - the ClassWizard will add and remove member functions here.
   //    DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
```

```
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_MAINFRM_H__755A532D_2BDA_11D4_8C84_000086319872__INCLUDED_)
```

```
// MainFrm.cpp : implementation of the CMainFrame class
//

#include "stdafx.h"
#include "noerase.h"
#include "MainFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CMainFrame

IMPLEMENT_DYNAMIC(CMainFrame, CMDIFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CMDIFrameWnd)
//{{AFX_MSG_MAP(CMainFrame)
   // NOTE - the ClassWizard will add and remove mapping macros here.
   //    DO NOT EDIT what you see in these blocks of generated code !
ON_WM_CREATE()
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

static UINT indicators[] =
{
ID_SEPARATOR,          // status line indicator
ID_INDICATOR_CAPS,
ID_INDICATOR_NUM,
ID_INDICATOR_SCRL,
};

/////////////////////////////////////////////////////////////////////////////
// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
// TODO: add member initialization code here
}

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
if (CMDIFrameWnd::OnCreate(lpCreateStruct) == -1)
   return -1;

if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD | WS_VISIBLE |
CBRS_TOP
   | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY |
CBRS_SIZE_DYNAMIC) ||
   !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
{
   TRACE0("Failed to create toolbar\n");
   return -1;      // fail to create
}
```

```
    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
          sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1;      // fail to create
    }

    // TODO: Delete these three lines if you don't want the toolbar to
    //  be dockable
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

    return 0;
    }

    BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
    {
    if( !CMDIFrameWnd::PreCreateWindow(cs) )
        return FALSE;
    // TODO: Modify the Window class or styles here by modifying
    //  the CREATESTRUCT cs

    return TRUE;
    }

    /////////////////////////////////////////////////////////////////////////////
    // CMainFrame diagnostics

    #ifdef _DEBUG
    void CMainFrame::AssertValid() const
    {
    CMDIFrameWnd::AssertValid();
    }

    void CMainFrame::Dump(CDumpContext& dc) const
    {
    CMDIFrameWnd::Dump(dc);
    }

    #endif //_DEBUG

    /////////////////////////////////////////////////////////////////////////////
    // CMainFrame message handlers
```

```
// ChildFrm.h : interface of the CChildFrame class
//
/////////////////////////////////////////////////////////////////////////////

#if
!defined(AFX_CHILDFRM_H__755A532F_2BDA_11D4_8C84_000086319872__I
NCLUDED_)
#define
AFX_CHILDFRM_H__755A532F_2BDA_11D4_8C84_000086319872__INCLUD
ED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CChildFrame : public CMDIChildWnd
{
DECLARE_DYNCREATE(CChildFrame)
public:
CChildFrame();

// Attributes
public:

// Operations
public:

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CChildFrame)
virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
//}}AFX_VIRTUAL

// Implementation
public:
virtual ~CChildFrame();
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif

// Generated message map functions
protected:
//{{AFX_MSG(CChildFrame)
   // NOTE - the ClassWizard will add and remove member functions here.
   //    DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the
previous line.

#endif //
!defined(AFX_CHILDFRM_H__755A532F_2BDA_11D4_8C84_000086319872__I
NCLUDED_)
```

```
// ChildFrm.cpp : implementation of the CChildFrame class
//

#include "stdafx.h"
#include "noerase.h"

#include "ChildFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CChildFrame

IMPLEMENT_DYNCREATE(CChildFrame, CMDIChildWnd)

BEGIN_MESSAGE_MAP(CChildFrame, CMDIChildWnd)
//{{AFX_MSG_MAP(CChildFrame)
   // NOTE - the ClassWizard will add and remove mapping macros here.
   //    DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CChildFrame construction/destruction

CChildFrame::CChildFrame()
{
// TODO: add member initialization code here
}

CChildFrame::~CChildFrame()
{
}

BOOL CChildFrame::PreCreateWindow(CREATESTRUCT& cs)
{
// TODO: Modify the Window class or styles here by modifying
//  the CREATESTRUCT cs

if( !CMDIChildWnd::PreCreateWindow(cs) )
   return FALSE;

return TRUE;
}

/////////////////////////////////////////////////////////////////////////////
// CChildFrame diagnostics

#ifdef _DEBUG
void CChildFrame::AssertValid() const
{
CMDIChildWnd::AssertValid();
}

void CChildFrame::Dump(CDumpContext& dc) const
{
```

```
CMDIChildWnd::Dump(dc);
}

#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////////
// CChildFrame message handlers
```

```
// noerase.h : main header file for the NOERASE application
//

#if
!defined(AFX_NOERASE_H__755A5329_2BDA_11D4_8C84_000086319872__I
NCLUDED_)
#define
AFX_NOERASE_H__755A5329_2BDA_11D4_8C84_000086319872__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#ifndef __AFXWIN_H__
#error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"       // main symbols

/////////////////////////////////////////////////////////////////////////////
// CNoeraseApp:
// See noerase.cpp for the implementation of this class
//

class CNoeraseApp : public CWinApp
{
public:
CNoeraseApp();

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CNoeraseApp)
public:
virtual BOOL InitInstance();
//}}AFX_VIRTUAL

// Implementation
//{{AFX_MSG(CNoeraseApp)
afx_msg void OnAppAbout();
    // NOTE - the ClassWizard will add and remove member functions here.
    //    DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the
previous line.

#endif //
!defined(AFX_NOERASE_H__755A5329_2BDA_11D4_8C84_000086319872__I
NCLUDED_)
```

```
// noerase.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "noerase.h"

#include "MainFrm.h"
#include "ChildFrm.h"
#include "noeraseDoc.h"
#include "noeraseView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CNoeraseApp

BEGIN_MESSAGE_MAP(CNoeraseApp, CWinApp)
//{{AFX_MSG_MAP(CNoeraseApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
    // NOTE - the ClassWizard will add and remove mapping macros here.
    //    DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard file based document commands
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
// Standard print setup command
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CNoeraseApp construction

CNoeraseApp::CNoeraseApp()
{
// TODO: add construction code here,
// Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////////////////
// The one and only CNoeraseApp object

CNoeraseApp theApp;

/////////////////////////////////////////////////////////////////////////////
// CNoeraseApp initialization

BOOL CNoeraseApp::InitInstance()
{
AfxEnableControlContainer();

// Standard initialization
// If you are not using these features and wish to reduce the size
//  of your final executable, you should remove from the following
//  the specific initialization routines you do not need.

#ifdef _AFXDLL
```

```
    Enable3dControls();         // Call this when using MFC in a shared DLL
    #else
    Enable3dControlsStatic(); // Call this when linking to MFC statically
    #endif

    // Change the registry key under which our settings are stored.
    // TODO: You should modify this string to be something appropriate
    // such as the name of your company or organization.
    SetRegistryKey(_T("Local AppWizard-Generated Applications"));

    LoadStdProfileSettings();  // Load standard INI file options (including MRU)

    // Register the application's document templates.  Document templates
    //  serve as the connection between documents, frame windows and views.

    CMultiDocTemplate* pDocTemplate;
    pDocTemplate = new CMultiDocTemplate(
        IDR_NOERASTYPE,
        RUNTIME_CLASS(CNoeraseDoc),
        RUNTIME_CLASS(CChildFrame), // custom MDI child frame
        RUNTIME_CLASS(CNoeraseView));
    AddDocTemplate(pDocTemplate);

    // create main MDI Frame window
    CMainFrame* pMainFrame = new CMainFrame;
    if (!pMainFrame->LoadFrame(IDR_MAINFRAME))
        return FALSE;
    m_pMainWnd = pMainFrame;

    // Parse command line for standard shell commands, DDE, file open
    CCommandLineInfo cmdInfo;
    ParseCommandLine(cmdInfo);

    // Dispatch commands specified on the command line
    if (!ProcessShellCommand(cmdInfo))
        return FALSE;

    // The main window has been initialized, so show and update it.
    pMainFrame->ShowWindow(m_nCmdShow);
    pMainFrame->UpdateWindow();

    return TRUE;
    }

    /////////////////////////////////////////////////////////////////////////////
    // CAboutDlg dialog used for App About

    class CAboutDlg : public CDialog
    {
    public:
    CAboutDlg();

    // Dialog Data
    //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CAboutDlg)
    protected:
```

```
virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
   // No message handlers
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
CDialog::DoDataExchange(pDX);
//{{AFX_DATA_MAP(CAboutDlg)
//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
//{{AFX_MSG_MAP(CAboutDlg)
   // No message handlers
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

// App command to run the dialog
void CNoeraseApp::OnAppAbout()
{
CAboutDlg aboutDlg;
aboutDlg.DoModal();
}

/////////////////////////////////////////////////////////////////////////////
// CNoeraseApp message handlers
```

```
// noeraseDoc.h : interface of the CNoeraseDoc class
//
/////////////////////////////////////////////////////////////////////////////

#if
!defined(AFX_NOERASEDOC_H__755A5331_2BDA_11D4_8C84_00008631987
2__INCLUDED_)
#define
AFX_NOERASEDOC_H__755A5331_2BDA_11D4_8C84_000086319872__INCL
UDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CNoeraseDoc : public CDocument
{
protected: // create from serialization only
CNoeraseDoc();
DECLARE_DYNCREATE(CNoeraseDoc)

// Attributes
public:

// Operations
public:

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CNoeraseDoc)
public:
virtual BOOL OnNewDocument();
virtual void Serialize(CArchive& ar);
//}}AFX_VIRTUAL

// Implementation
public:
virtual ~CNoeraseDoc();
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
//{{AFX_MSG(CNoeraseDoc)
    // NOTE - the ClassWizard will add and remove member functions here.
    //    DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

/////////////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the
previous line.
```

```
#endif //
!defined(AFX_NOERASEDOC_H__755A5331_2BDA_11D4_8C84_00008631987
2__INCLUDED_)
```

```
// noeraseDoc.cpp : implementation of the CNoeraseDoc class
//

#include "stdafx.h"
#include "noerase.h"

#include "noeraseDoc.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CNoeraseDoc

IMPLEMENT_DYNCREATE(CNoeraseDoc, CDocument)

BEGIN_MESSAGE_MAP(CNoeraseDoc, CDocument)
//{{AFX_MSG_MAP(CNoeraseDoc)
   // NOTE - the ClassWizard will add and remove mapping macros here.
   //    DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CNoeraseDoc construction/destruction

CNoeraseDoc::CNoeraseDoc()
{
// TODO: add one-time construction code here
}

CNoeraseDoc::~CNoeraseDoc()
{
}

BOOL CNoeraseDoc::OnNewDocument()
{
if (!CDocument::OnNewDocument())
   return FALSE;

// TODO: add reinitialization code here
// (SDI documents will reuse this document)

return TRUE;
}

/////////////////////////////////////////////////////////////////////////////
// CNoeraseDoc serialization

void CNoeraseDoc::Serialize(CArchive& ar)
{
if (ar.IsStoring())
{
   // TODO: add storing code here
}
else
{
```

```
        // TODO: add loading code here
    }
}

/////////////////////////////////////////////////////////////////////////////
// CNoeraseDoc diagnostics

#ifdef _DEBUG
void CNoeraseDoc::AssertValid() const
{
CDocument::AssertValid();
}

void CNoeraseDoc::Dump(CDumpContext& dc) const
{
CDocument::Dump(dc);
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////////
// CNoeraseDoc commands
```

```
// noeraseView.h : interface of the CNoeraseView class
//
/////////////////////////////////////////////////////////////////////////////

#if
!defined(AFX_NOERASEVIEW_H__755A5333_2BDA_11D4_8C84_0000863198
72__INCLUDED_)
#define
AFX_NOERASEVIEW_H__755A5333_2BDA_11D4_8C84_000086319872__INC
LUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

class CNoeraseView : public CView
{
protected: // create from serialization only
CNoeraseView();
DECLARE_DYNCREATE(CNoeraseView)

// Attributes
public:
CNoeraseDoc* GetDocument();

// Operations
public:

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CNoeraseView)
public:
virtual void OnDraw(CDC* pDC);  // overridden to draw this view
virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
//}}AFX_VIRTUAL

// Implementation
public:
virtual ~CNoeraseView();
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
//{{AFX_MSG(CNoeraseView)
afx_msg BOOL OnEraseBkgnd(CDC* pDC);
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

#ifndef _DEBUG  // debug version in noeraseView.cpp
inline CNoeraseDoc* CNoeraseView::GetDocument()
```

```
    { return (CNoeraseDoc*)m_pDocument; }
#endif

//////////////////////////////////////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the
previous line.

#endif //
!defined(AFX_NOERASEVIEW_H__755A5333_2BDA_11D4_8C84_0000863198
72__INCLUDED_)
```

```
// noeraseView.cpp : implementation of the CNoeraseView class
//

#include "stdafx.h"
#include "noerase.h"

#include "noeraseDoc.h"
#include "noeraseView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CNoeraseView

IMPLEMENT_DYNCREATE(CNoeraseView, CView)

BEGIN_MESSAGE_MAP(CNoeraseView, CView)
//{{AFX_MSG_MAP(CNoeraseView)
ON_WM_ERASEBKGND()
//}}AFX_MSG_MAP
// Standard printing commands
ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_DIRECT, CView::OnFilePrint)
ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CNoeraseView construction/destruction

CNoeraseView::CNoeraseView()
{
// TODO: add construction code here
}

CNoeraseView::~CNoeraseView()
{
}

BOOL CNoeraseView::PreCreateWindow(CREATESTRUCT& cs)
{
// TODO: Modify the Window class or styles here by modifying
//  the CREATESTRUCT cs

return CView::PreCreateWindow(cs);
}

/////////////////////////////////////////////////////////////////////////////
// CNoeraseView drawing

void CNoeraseView::OnDraw(CDC* pDC)
{
CNoeraseDoc* pDoc = GetDocument();
ASSERT_VALID(pDoc);
// TODO: add draw code for native data here
}
```

```
////////////////////////////////////////////////////////////////////////////
// CNoeraseView printing

BOOL CNoeraseView::OnPreparePrinting(CPrintInfo* pInfo)
{
// default preparation
return DoPreparePrinting(pInfo);
}

void CNoeraseView::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
// TODO: add extra initialization before printing
}

void CNoeraseView::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
// TODO: add cleanup after printing
}

////////////////////////////////////////////////////////////////////////////
// CNoeraseView diagnostics

#ifdef _DEBUG
void CNoeraseView::AssertValid() const
{
CView::AssertValid();
}

void CNoeraseView::Dump(CDumpContext& dc) const
{
CView::Dump(dc);
}

CNoeraseDoc* CNoeraseView::GetDocument() // non-debug version is inline
{
ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CNoeraseDoc)));
return (CNoeraseDoc*)m_pDocument;
}
#endif //_DEBUG

////////////////////////////////////////////////////////////////////////////
// CNoeraseView message handlers

BOOL CNoeraseView::OnEraseBkgnd(CDC* pDC)
{
return true;
}
```

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by NOERASE.RC
//
#define IDD_ABOUTBOX                100
#define IDR_MAINFRAME               128
#define IDR_NOERASTYPE              129

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_3D_CONTROLS            1
#define _APS_NEXT_RESOURCE_VALUE    130
#define _APS_NEXT_CONTROL_VALUE     1000
#define _APS_NEXT_SYMED_VALUE       101
#define _APS_NEXT_COMMAND_VALUE     32771
#endif
#endif
```

```
// stdafx.h : include file for standard system include files,
//  or project specific include files that are used frequently, but
//      are changed infrequently
//

#if
!defined(AFX_STDAFX_H__755A532B_2BDA_11D4_8C84_000086319872__IN
CLUDED_)
#define
AFX_STDAFX_H__755A532B_2BDA_11D4_8C84_000086319872__INCLUDED
_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#define VC_EXTRALEAN        // Exclude rarely-used stuff from Windows headers

#include <afxwin.h>         // MFC core and standard components
#include <afxext.h>         // MFC extensions
#include <afxdisp.h>        // MFC Automation classes
#include <afxdtctl.h>       // MFC support for Internet Explorer 4 Common Controls
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h>         // MFC support for Windows Common Controls
#endif // _AFX_NO_AFXCMN_SUPPORT

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the
previous line.

#endif //
!defined(AFX_STDAFX_H__755A532B_2BDA_11D4_8C84_000086319872__IN
CLUDED_)


// stdafx.cpp : source file that includes just the standard includes
//  noerase.pch will be the pre-compiled header
//  stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```

```
//Microsoft Visual C++ generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//

#include "afxres.h"

//////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

#ifdef APSTUDIO_INVOKED

//////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
"resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
"#include ""afxres.h""\r\n"
"\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
   "#define _AFX_NO_SPLITTER_RESOURCES\r\n"
   "#define _AFX_NO_OLE_RESOURCES\r\n"
   "#define _AFX_NO_TRACKER_RESOURCES\r\n"
   "#define _AFX_NO_PROPERTY_RESOURCES\r\n"
"\r\n"
"#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)\r\n"
"#ifdef _WIN32\r\n"
"LANGUAGE 9, 1\r\n"
"#pragma code_page(1252)\r\n"
"#endif //_WIN32\r\n"
"#include ""res\\noerase.rc2""  // non-Microsoft Visual C++ edited resources\r\n"
"#include ""afxres.rc""      // Standard components\r\n"
"#include ""afxprint.rc""    // printing/print preview resources\r\n"
"#endif\r\n"
"\0"
END

#endif    // APSTUDIO_INVOKED

//////////////////////////////////////////////////////////////////////////////
//
// Icon
//

// Icon with lowest ID value placed first to ensure application icon
```

```
// remains consistent on all systems.

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE 9, 1
#pragma code_page(1252)
#endif // _WIN32
IDR_MAINFRAME       ICON   DISCARDABLE    "res\\noerase.ico"
IDR_NOERASTYPE      ICON   DISCARDABLE    "res\\noeraseDoc.ico"
#endif

/////////////////////////////////////////////////////////////////////////////
//
// Bitmap
//

IDR_MAINFRAME       BITMAP  MOVEABLE PURE   "res\\Toolbar.bmp"

/////////////////////////////////////////////////////////////////////////////
//
// Toolbar
//

IDR_MAINFRAME TOOLBAR DISCARDABLE  16, 15
BEGIN
    BUTTON      ID_FILE_NEW
    BUTTON      ID_FILE_OPEN
    BUTTON      ID_FILE_SAVE
SEPARATOR
    BUTTON      ID_EDIT_CUT
    BUTTON      ID_EDIT_COPY
    BUTTON      ID_EDIT_PASTE
SEPARATOR
    BUTTON      ID_FILE_PRINT
SEPARATOR
    BUTTON      ID_APP_ABOUT
END

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE 9, 1
#pragma code_page(1252)
#endif // _WIN32
/////////////////////////////////////////////////////////////////////////////
//
// Menu
//

IDR_MAINFRAME MENU PRELOAD DISCARDABLE
BEGIN
POPUP "&File"
BEGIN
    MENUITEM "&New\tCtrl+N",            ID_FILE_NEW
    MENUITEM "&Open...\tCtrl+O",        ID_FILE_OPEN
    MENUITEM SEPARATOR
    MENUITEM "P&rint Setup...",         ID_FILE_PRINT_SETUP
    MENUITEM SEPARATOR
    MENUITEM "Recent File",             ID_FILE_MRU_FILE1,GRAYED
    MENUITEM SEPARATOR
    MENUITEM "E&xit",                   ID_APP_EXIT
```

```
    END
    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",                ID_VIEW_TOOLBAR
        MENUITEM "&Status Bar",             ID_VIEW_STATUS_BAR
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&About noerase...",       ID_APP_ABOUT
    END
    END
    IDR_NOERASTYPE MENU PRELOAD DISCARDABLE
    BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",            ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",        ID_FILE_OPEN
        MENUITEM "&Close",                  ID_FILE_CLOSE
        MENUITEM "&Save\tCtrl+S",           ID_FILE_SAVE
        MENUITEM "Save &As...",             ID_FILE_SAVE_AS
        MENUITEM SEPARATOR
        MENUITEM "&Print...\tCtrl+P",       ID_FILE_PRINT
        MENUITEM "Print Pre&view",          ID_FILE_PRINT_PREVIEW
        MENUITEM "P&rint Setup...",         ID_FILE_PRINT_SETUP
        MENUITEM SEPARATOR
        MENUITEM "Recent File",             ID_FILE_MRU_FILE1,GRAYED
        MENUITEM SEPARATOR
        MENUITEM "E&xit",                   ID_APP_EXIT
    END
    POPUP "&Edit"
    BEGIN
        MENUITEM "&Undo\tCtrl+Z",           ID_EDIT_UNDO
        MENUITEM SEPARATOR
        MENUITEM "Cu&t\tCtrl+X",            ID_EDIT_CUT
        MENUITEM "&Copy\tCtrl+C",           ID_EDIT_COPY
        MENUITEM "&Paste\tCtrl+V",          ID_EDIT_PASTE
    END
    POPUP "&View"
    BEGIN
        MENUITEM "&Toolbar",                ID_VIEW_TOOLBAR
        MENUITEM "&Status Bar",             ID_VIEW_STATUS_BAR
    END
    POPUP "&Window"
    BEGIN
        MENUITEM "&New Window",             ID_WINDOW_NEW
        MENUITEM "&Cascade",                ID_WINDOW_CASCADE
        MENUITEM "&Tile",                   ID_WINDOW_TILE_HORZ
        MENUITEM "&Arrange Icons",          ID_WINDOW_ARRANGE
    END
    POPUP "&Help"
    BEGIN
        MENUITEM "&About noerase...",       ID_APP_ ABOUT
    END
    END

    //////////////////////////////////////////////////////////////////////
    //
    // Accelerator
    //
```

```
IDR_MAINFRAME ACCELERATORS PRELOAD MOVEABLE PURE
BEGIN
"N",        ID_FILE_NEW,        VIRTKEY,CONTROL
"O",        ID_FILE_OPEN,       VIRTKEY,CONTROL
"S",        ID_FILE_SAVE,       VIRTKEY,CONTROL
"P",        ID_FILE_PRINT,      VIRTKEY,CONTROL
"Z",        ID_EDIT_UNDO,       VIRTKEY,CONTROL
"X",        ID_EDIT_CUT,        VIRTKEY,CONTROL
"C",        ID_EDIT_COPY,       VIRTKEY,CONTROL
"V",        ID_EDIT_PASTE,      VIRTKEY,CONTROL
VK_BACK,    ID_EDIT_UNDO,       VIRTKEY,ALT
VK_DELETE,  ID_EDIT_CUT,        VIRTKEY,SHIFT
VK_INSERT,  ID_EDIT_COPY,       VIRTKEY,CONTROL
VK_INSERT,  ID_EDIT_PASTE,      VIRTKEY,SHIFT
VK_F6,      ID_NEXT_PANE,       VIRTKEY
VK_F6,      ID_PREV_PANE,       VIRTKEY,SHIFT
END

/////////////////////////////////////////////////////////////////////////////
//
// Dialog
//

IDD_ABOUTBOX DIALOG DISCARDABLE  0, 0, 235, 55
CAPTION "About noerase"
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
FONT 8, "MS Sans Serif"
BEGIN
ICON        IDR_MAINFRAME,IDC_STATIC,11,17,20,20
LTEXT       "noerase Version 1.0",IDC_STATIC,40,10,119,8,
            SS_NOPREFIX
LTEXT       "Copyright (C) 2000",IDC_STATIC,40,25,119,8
DEFPUSHBUTTON  "OK",IDOK,178,7,50,14,WS_GROUP
END

/////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO    VERSIONINFO
 FILEVERSION      1,0,0,1
 PRODUCTVERSION   1,0,0,1
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x1L
 FILESUBTYPE 0x0L
BEGIN
BLOCK "StringFileInfo"
BEGIN
    BLOCK "040904B0"
   BEGIN
      VALUE "CompanyName",     "\0"
      VALUE "FileDescription", "noerase MFC Application\0"
      VALUE "FileVersion",     "1, 0, 0, 1\0"
```

```
            VALUE "InternalName",   "noerase\0"
            VALUE "LegalCopyright", "Copyright (C) 2000\0"
            VALUE "LegalTrademarks", "\0"
            VALUE "OriginalFilename","noerase.EXE\0"
            VALUE "ProductName",    "noerase Application\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_ABOUTBOX, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 228
        TOPMARGIN, 7
        BOTTOMMARGIN, 48
    END
END
#endif    // APSTUDIO_INVOKED

////////////////////////////////////////////////////////////////////////////
//
// String Table
//

STRINGTABLE PRELOAD DISCARDABLE
BEGIN
IDR_MAINFRAME       "noerase"
IDR_NOERASTYPE      "\nNoeras\nNoeras\n\n\nNoerase.Document\nNoeras
Document"
END
STRINGTABLE PRELOAD DISCARDABLE
BEGIN
AFX_IDS_APP_TITLE     "noerase"
AFX_IDS_IDLEMESSAGE   "Ready"
END
STRINGTABLE DISCARDABLE
BEGIN
ID_INDICATOR_EXT      "EXT"
ID_INDICATOR_CAPS     "CAP"
ID_INDICATOR_NUM      "NUM"
ID_INDICATOR_SCRL     "SCRL"
ID_INDICATOR_OVR      "OVR"
ID_INDICATOR_REC      "REC"
END
STRINGTABLE DISCARDABLE
BEGIN
ID_FILE_NEW         "Create a new document\nNew"
```

```
    ID_FILE_OPEN        "Open an existing document\nOpen"
    ID_FILE_CLOSE       "Close the active document\nClose"
    ID_FILE_SAVE        "Save the active document\nSave"
    ID_FILE_SAVE_AS     "Save the active document with a new name\nSave As"
    ID_FILE_PAGE_SETUP  "Change the printing options\nPage Setup"
    ID_FILE_PRINT_SETUP "Change the printer and printing options\nPrint Setup"
    ID_FILE_PRINT       "Print the active document\nPrint"
    ID_FILE_PRINT_PREVIEW "Display full pages\nPrint Preview"
    ID_APP_ABOUT        "Display program information, version number and
copyright\nAbout"
    ID_APP_EXIT         "Quit the application; prompts to save documents\nExit"
    ID_FILE_MRU_FILE1   "Open this document"
    ID_FILE_MRU_FILE2   "Open this document"
    ID_FILE_MRU_FILE3   "Open this document"
    ID_FILE_MRU_FILE4   "Open this document"
    ID_FILE_MRU_FILE5   "Open this document"
    ID_FILE_MRU_FILE6   "Open this document"
    ID_FILE_MRU_FILE7   "Open this document"
    ID_FILE_MRU_FILE8   "Open this document"
    ID_FILE_MRU_FILE9   "Open this document"
    ID_FILE_MRU_FILE10  "Open this document"
    ID_FILE_MRU_FILE11  "Open this document"
    ID_FILE_MRU_FILE12  "Open this document"
    ID_FILE_MRU_FILE13  "Open this document"
    ID_FILE_MRU_FILE14  "Open this document"
    ID_FILE_MRU_FILE15  "Open this document"
    ID_FILE_MRU_FILE16  "Open this document"
    ID_NEXT_PANE        "Switch to the next window pane\nNext Pane"
    ID_PREV_PANE        "Switch back to the previous window pane\nPrevious
Pane"
    ID_WINDOW_NEW       "Open another window for the active document\nNew
Window"
    ID_WINDOW_ARRANGE   "Arrange icons at the bottom of the
window\nArrange Icons"
    ID_WINDOW_CASCADE   "Arrange windows so they overlap\nCascade
Windows"
    ID_WINDOW_TILE_HORZ "Arrange windows as non-overlapping tiles\nTile
Windows"
    ID_WINDOW_TILE_VERT "Arrange windows as non-overlapping tiles\nTile
Windows"
    ID_WINDOW_SPLIT     "Split the active window into panes\nSplit"
    ID_EDIT_CLEAR       "Erase the selection\nErase"
    ID_EDIT_CLEAR_ALL   "Erase everything\nErase All"
    ID_EDIT_COPY        "Copy the selection and put it on the Clipboard\nCopy"
    ID_EDIT_CUT         "Cut the selection and put it on the Clipboard\nCut"
    ID_EDIT_FIND        "Find the specified text\nFind"
    ID_EDIT_PASTE       "Insert Clipboard contents\nPaste"
    ID_EDIT_REPEAT      "Repeat the last action\nRepeat"
    ID_EDIT_REPLACE     "Replace specific text with different text\nReplace"
    ID_EDIT_SELECT_ALL  "Select the entire document\nSelect All"
    ID_EDIT_UNDO        "Undo the last action\nUndo"
    ID_EDIT_REDO        "Redo the previously undone action\nRedo"
    ID_VIEW_TOOLBAR     "Show or hide the toolbar\nToggle ToolBar"
    ID_VIEW_STATUS_BAR  "Show or hide the status bar\nToggle StatusBar"
    END

    STRINGTABLE DISCARDABLE
    BEGIN
    AFX_IDS_SCSIZE      "Change the window size"
    AFX_IDS_SCMOVE      "Change the window position"
```

```
    AFX_IDS_SCMINIMIZE      "Reduce the window to an icon"
    AFX_IDS_SCMAXIMIZE      "Enlarge the window to full size"
    AFX_IDS_SCNEXTWINDOW    "Switch to the next document window"
    AFX_IDS_SCPREVWINDOW    "Switch to the previous document window"
    AFX_IDS_SCCLOSE         "Close the active window and prompts to save the
    documents"
    AFX_IDS_SCRESTORE       "Restore the window to normal size"
    AFX_IDS_SCTASKLIST      "Activate Task List"
    AFX_IDS_MDICHILD        "Activate this window"
    AFX_IDS_PREVIEW_CLOSE   "Close print preview mode\nCancel Preview"
    END

    #endif

    #ifndef APSTUDIO_INVOKED
    ////////////////////////////////////////////////////////////////////////////
    //
    // Generated from the TEXTINCLUDE 3 resource.
    //

    #define _AFX_NO_SPLITTER_RESOURCES
    #define _AFX_NO_OLE_RESOURCES
    #define _AFX_NO_TRACKER_RESOURCES
    #define _AFX_NO_PROPERTY_RESOURCES

    #if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
    #ifdef _WIN32
    LANGUAGE 9, 1
    #pragma code_page(1252)
    #endif //_WIN32
    #include "res\\noerase.rc2"  // non-Microsoft Visual C++ edited resources
    #include "afxres.rc"        // Standard components
    #include "afxprint.rc"  // printing/print preview resources
    #endif

    #endif    // not APSTUDIO_INVOKED
```

CLAIMS

What is claimed is:

1. A method for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, comprising:

providing screen pixel data for rendering on a computer monitor, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin;

detecting that a second window is going to be displayed on the computer monitor;

determining the position and size of the second window;

determining, based on the position and size of the second window, a portion of the screen pixel data wherein the first window is going to be covered by the second window; and

replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

2. The method of claim 1 further comprising the step of registering an application to include a system-wide hook in order to monitor window events occurring within a windows operating system, and wherein said detecting comprises receiving notification of a window event from the windows operating system.

3. The method of claim 2 wherein the system-wide hook is a Windows CBT hook.

4. The method of claim 2 herein the system-wide hook is a Windows CallWndProc hook.

5. The method of claim 1 wherein said detecting detects that a new window is going to be opened.

6. The method of claim 1 wherein said detecting detects that an existing window is going to be enlarged.

7. The method of claim 1 wherein said detecting detects that an existing window is going to be maximized.

8. The method of claim 1 wherein said detecting detects that an existing window is going to be moved from behind the first window to in front of the first window.

9. The method of claim 1 wherein the substitute pixel data is white pixel data.

10. A system for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, comprising:

a computer monitor on which screen pixel data is rendered, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin;

an event detector detecting that a second window is going to be displayed on the computer monitor;

a window processor for determining the position and size of the second window, and for determining, based on the position and size of the second window, a portion of the screen pixel data wherein the first window is going to be covered by the second window; and

a pixel processor for replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

11. The system of claim 10 further comprising a hook registry for registering an application to include a system-wide hook in order to monitor window events occurring within a windows operating system, and wherein said event detector comprises a notification receiver for receiving notification of a window event from the windows operating system.

12. The system of claim 11 wherein the system-wide hook is a Windows CBT hook.

13. The system of claim 11 wherein the system-wide hook is a Windows CallWndProc hook.

14. The system of claim 10 wherein said event detector detects that a new window is going to be opened.

15. The system of claim 10 wherein said event detector detects that an existing window is going to be enlarged.

16. The system of claim 10 wherein said event detector detects that an existing window is going to be maximized.

17. The system of claim 10 wherein said event detector detects that that an existing window is going to be moved from behind the first window to in front of the first window.

18. The system of claim 10 wherein the substitute pixel data is white pixel data.

19. A method for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, comprising:

providing screen pixel data for rendering on a computer monitor, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin;

detecting that a second window is going to be displayed on the computer monitor;

determining the position and size of the second window;

determining, based on the position and size of the second window, a portion of the screen pixel data wherein the proprietary digital image data is going to be covered by the second window; and

replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

20. A system for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, comprising:

a computer monitor on which screen pixel data is rendered, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin;

an event detector detecting that a second window is going to be displayed on the computer monitor;

a window processor for determining the position and size of the second window, and for determining, based on the position and size of the second

window, a portion of the screen pixel data wherein the proprietary digital image data is going to be covered by the second window; and

a pixel processor for replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed.

ABSTRACT

A method for preventing copying of proprietary digital image data that is rendered within a window displayed on a computer monitor, including providing screen pixel data for rendering on a computer monitor, the screen pixel data including pixel data for a first window having proprietary digital image data therewithin, detecting that a second window is going to be displayed on the computer monitor, determining the position and size of the second window, determining, based on the position and size of the second window, a portion of the screen pixel data wherein the first window is going to be covered by the second window, and replacing the portion of the screen pixel data with substitute pixel data, prior to the second window being displayed. A system is also described and claimed.

000780 6629E960

FIGURE 1 (PRIOR ART)

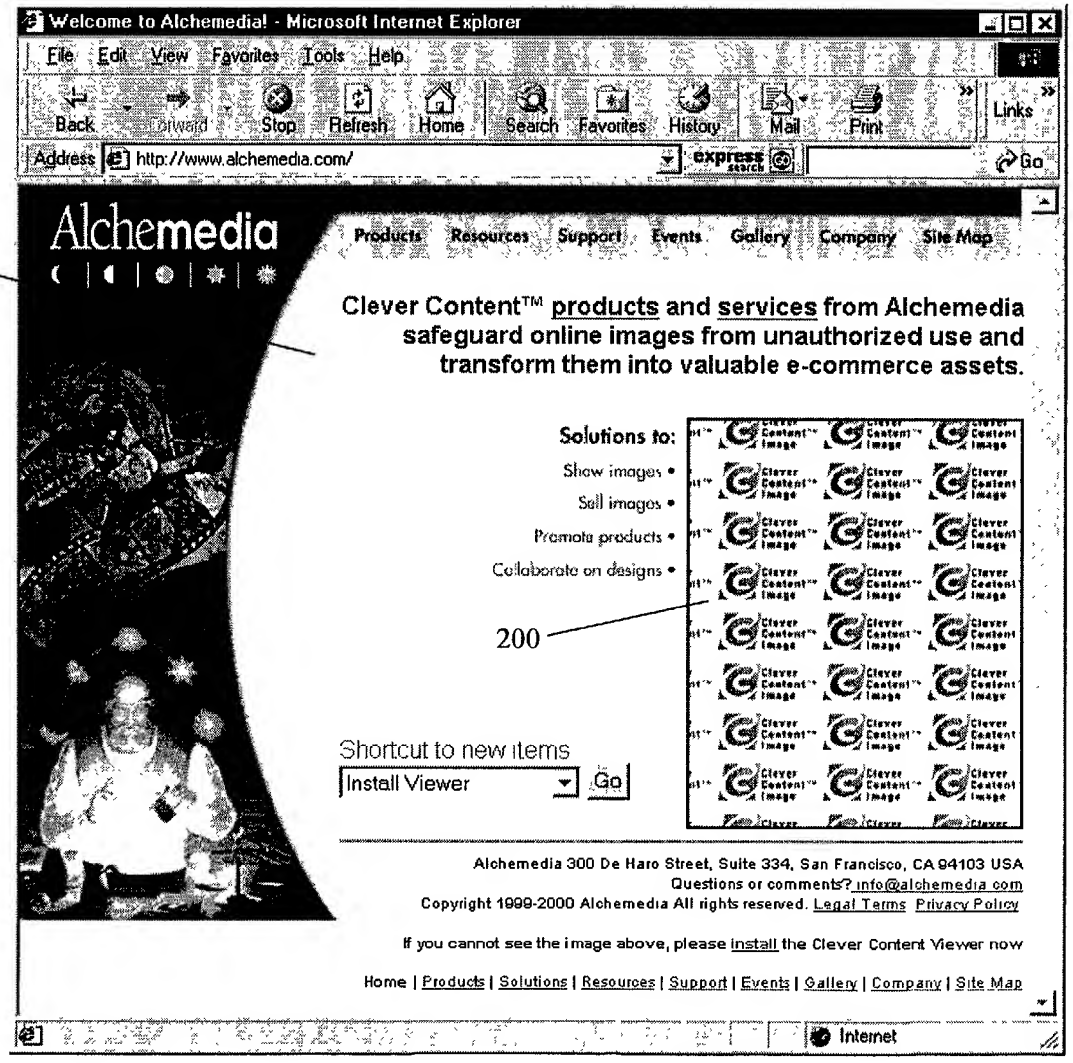


FIGURE 2 (PRIOR ART)

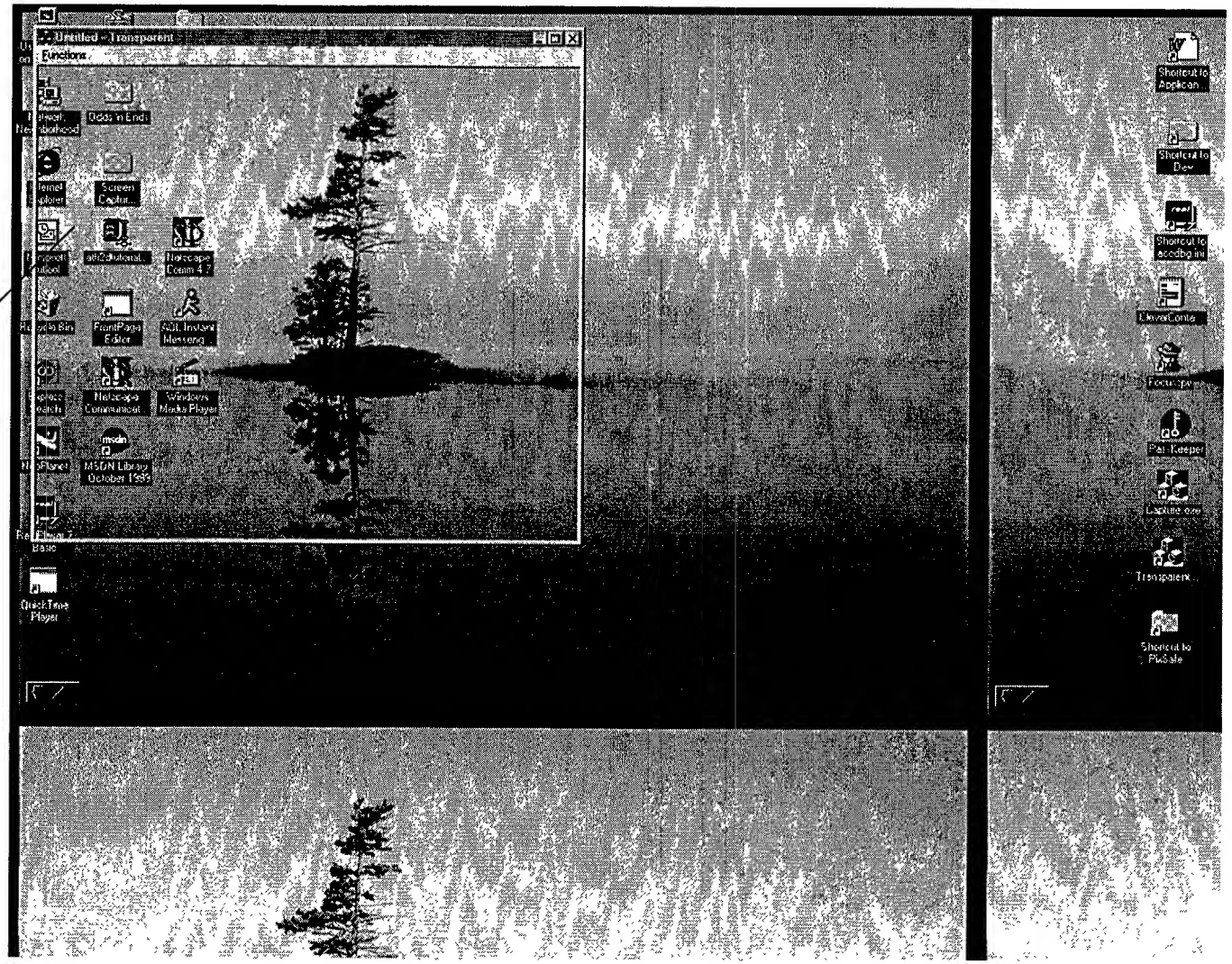


FIGURE 3 (PRIOR ART)

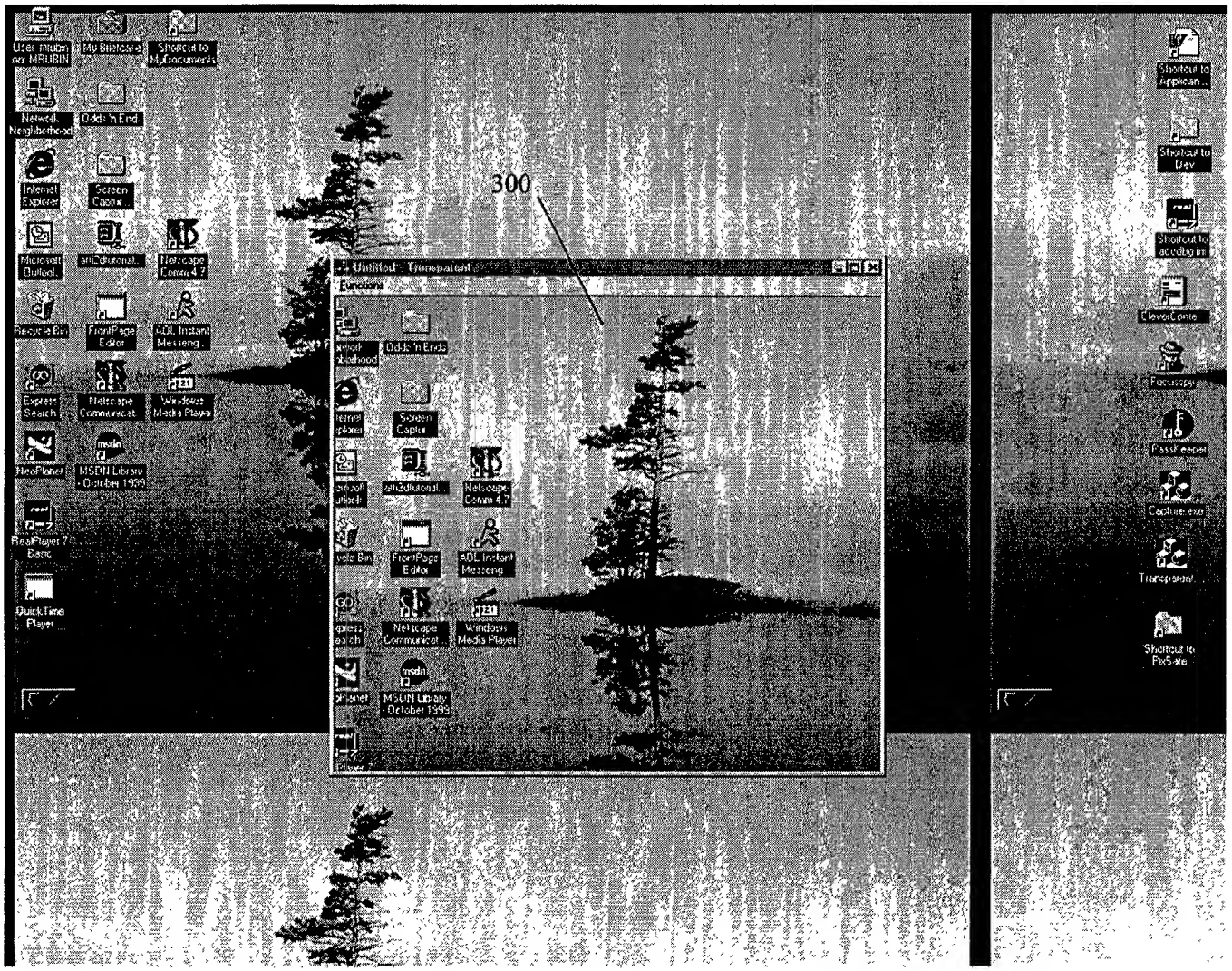


FIGURE 4 (PRIOR ART)

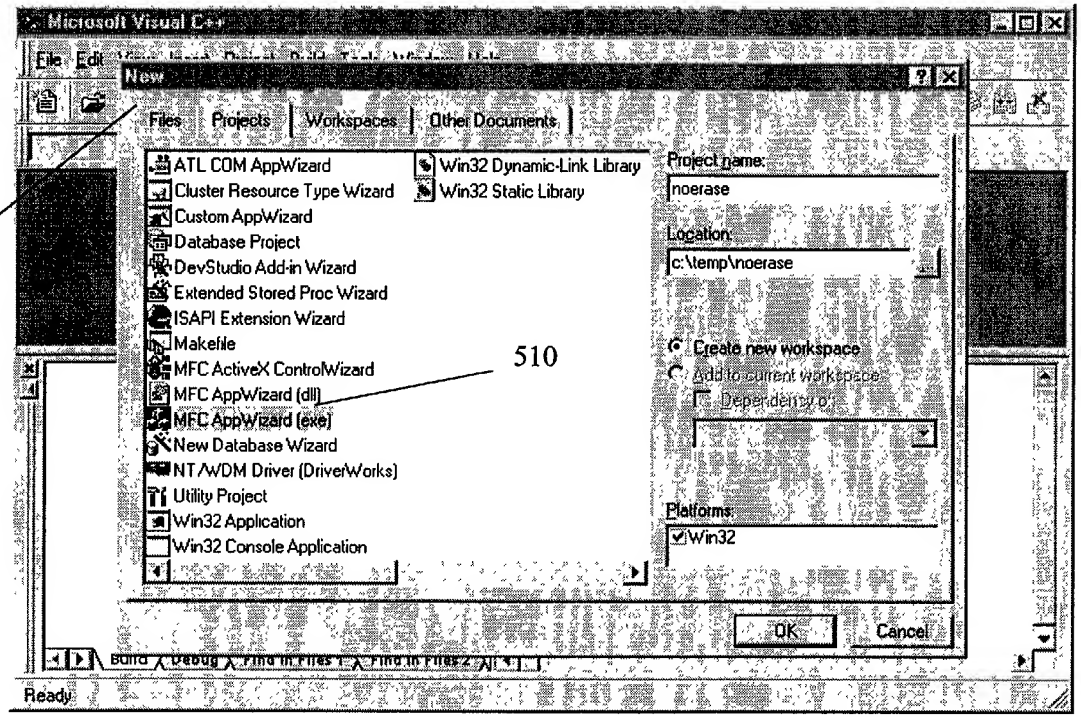


FIGURE 5A (PRIOR ART)

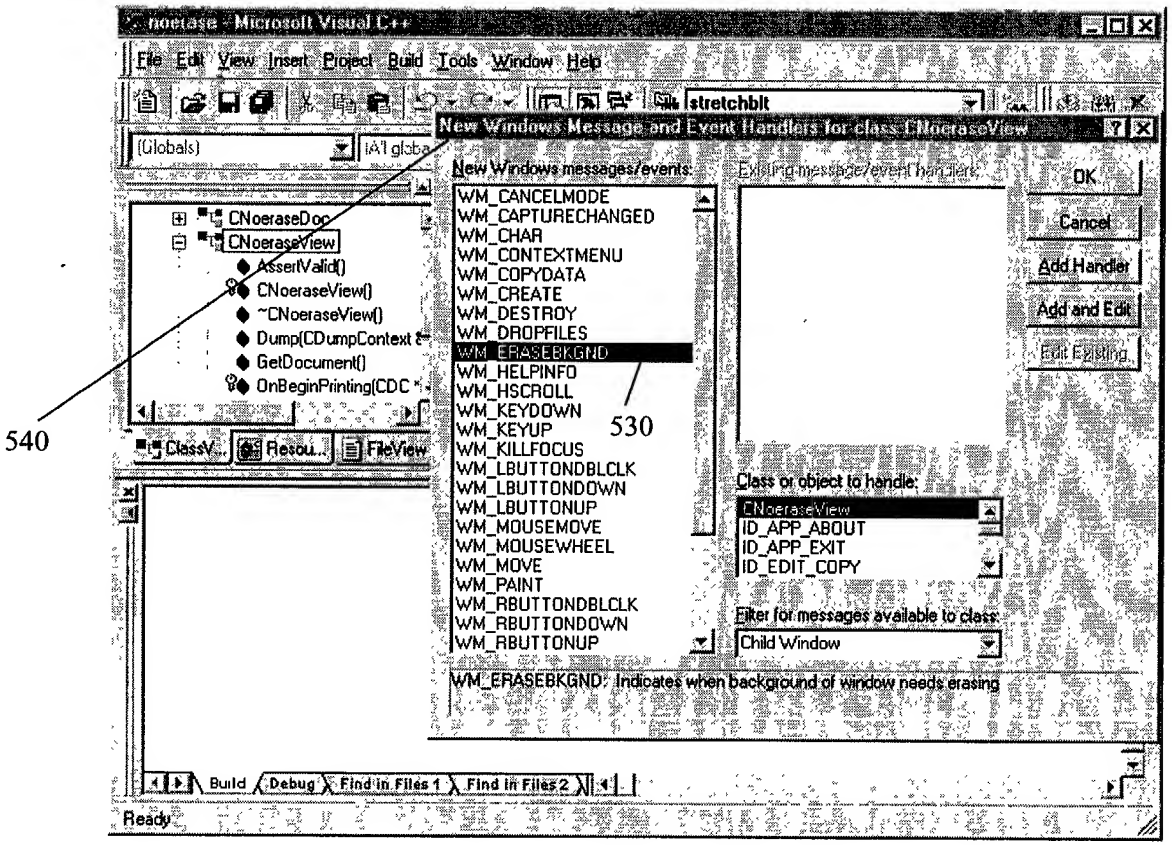
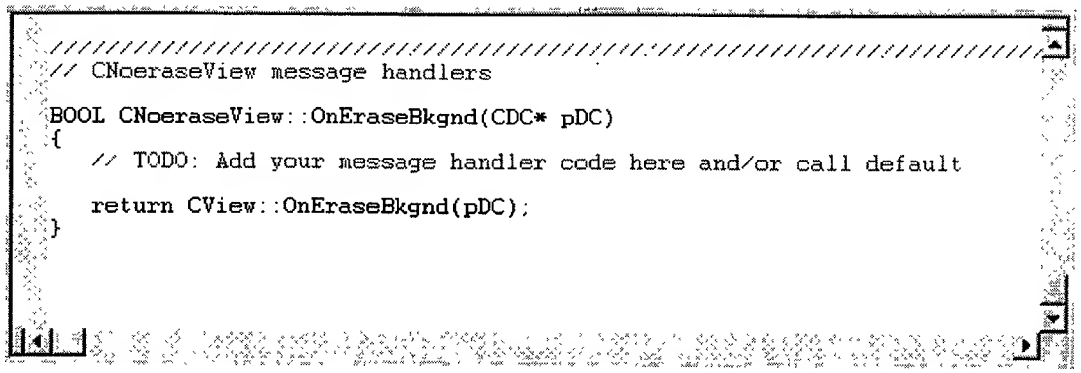


FIGURE 5B (PRIOR ART)

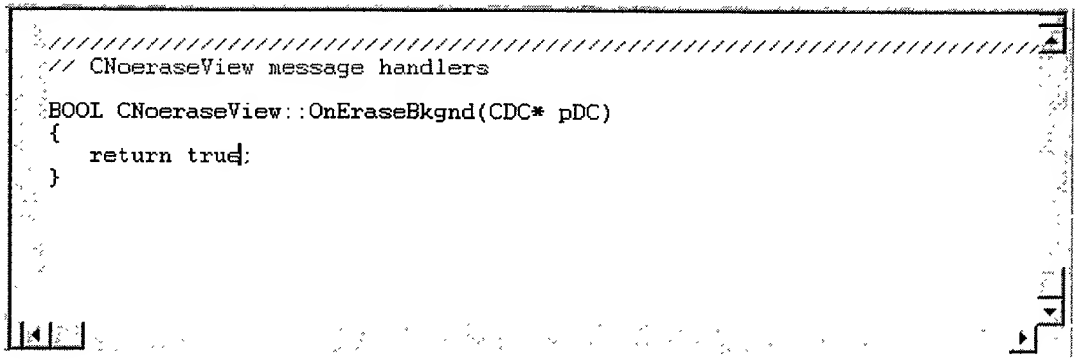


```
//////////////////////////////////////////////////////////////////////////
// CNoeraseView message handlers

BOOL CNoeraseView::OnEraseBkgnd(CDC* pDC)
{
    // TODO: Add your message handler code here and/or call default

    return CView::OnEraseBkgnd(pDC);
}
```

FIGURE 5C (PRIOR ART)



```
//////////////////////////////////////////////////////////////////////////
// CNoeraseView message handlers

BOOL CNoeraseView::OnEraseBkgnd(CDC* pDC)
{
    return true;
}
```

FIGURE 5D (PRIOR ART)

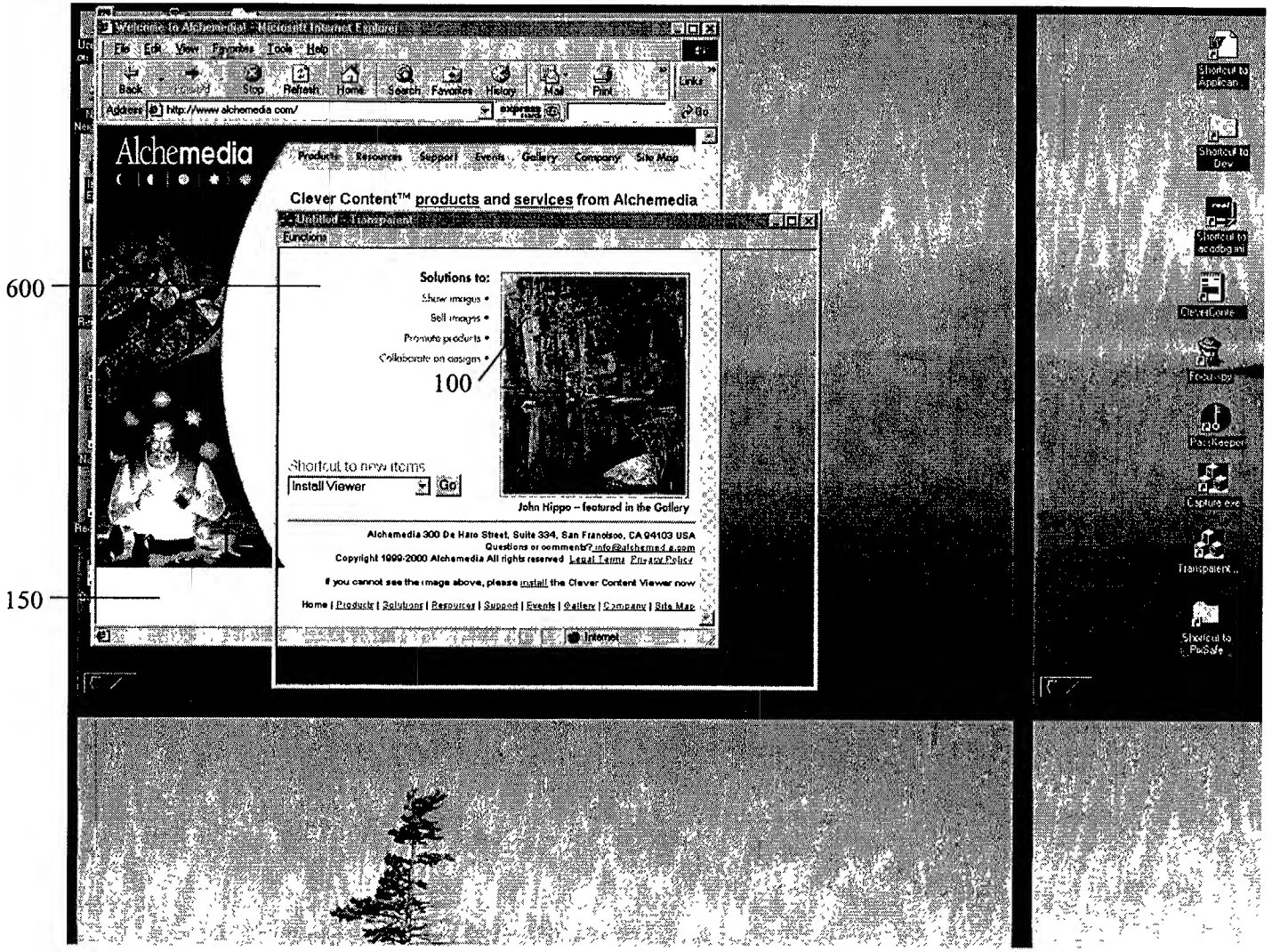


FIGURE 6 (PRIOR ART)

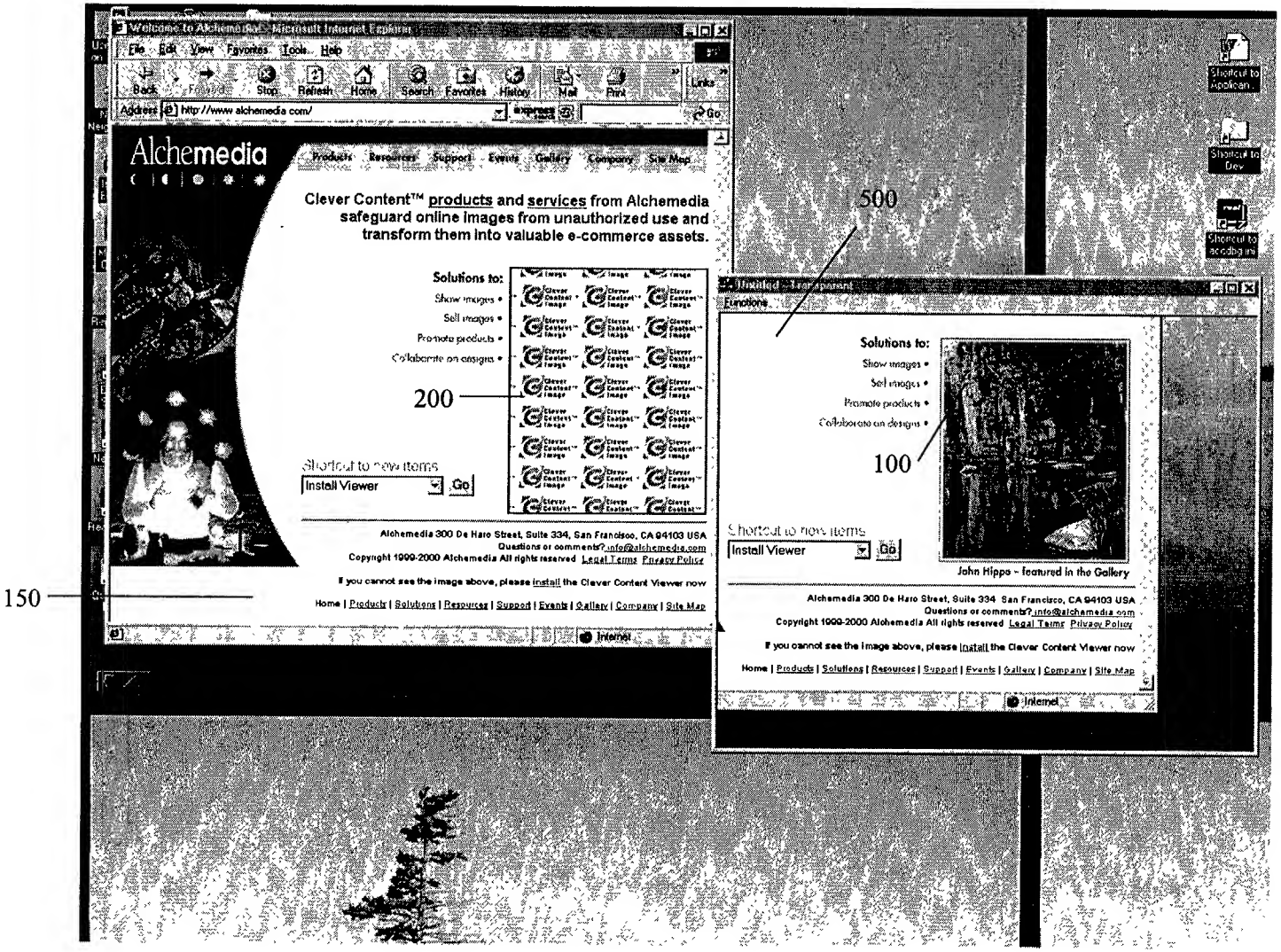


FIGURE 7 (PRIOR ART)

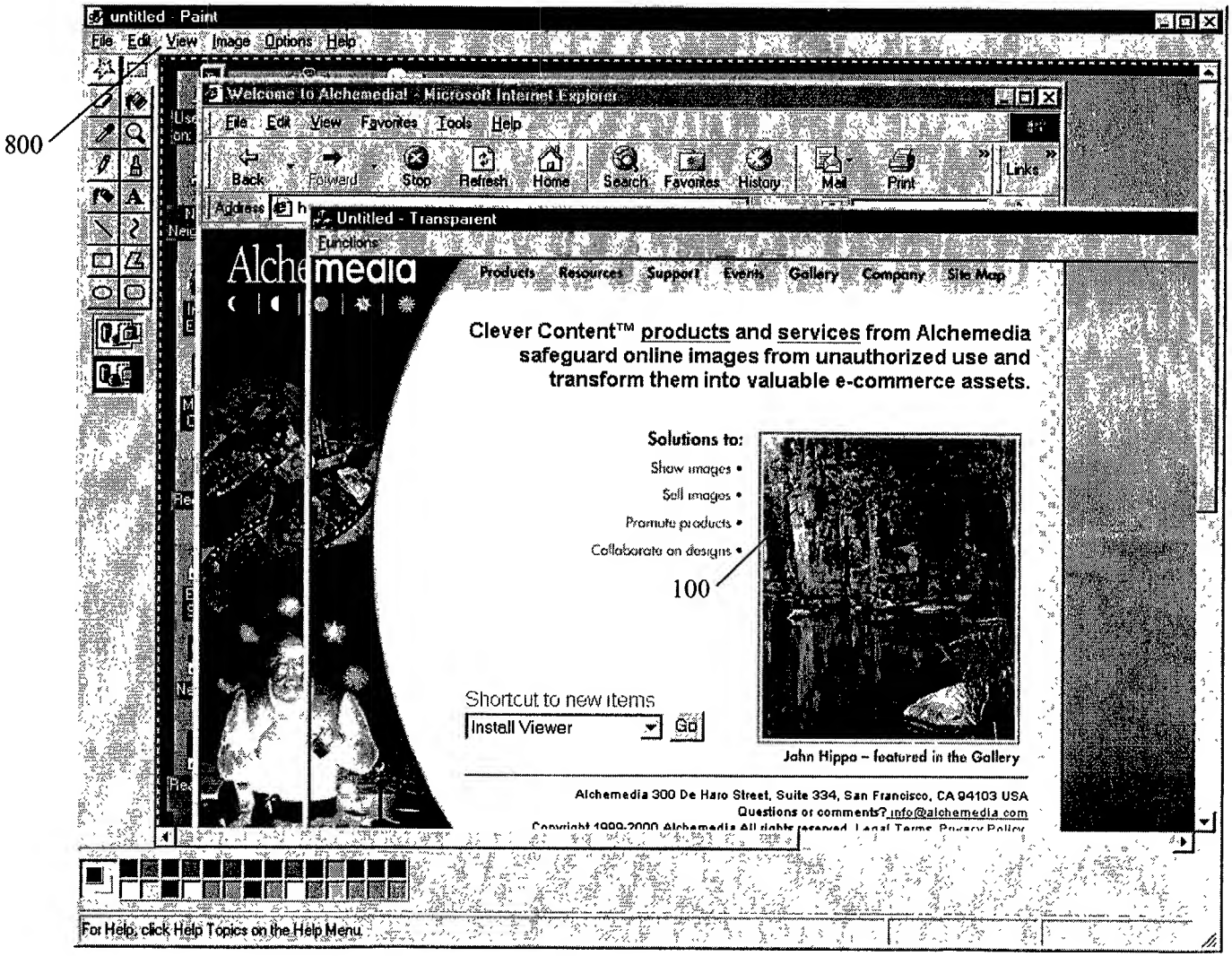


FIGURE 8 (PRIOR ART)

000130 "6629E960

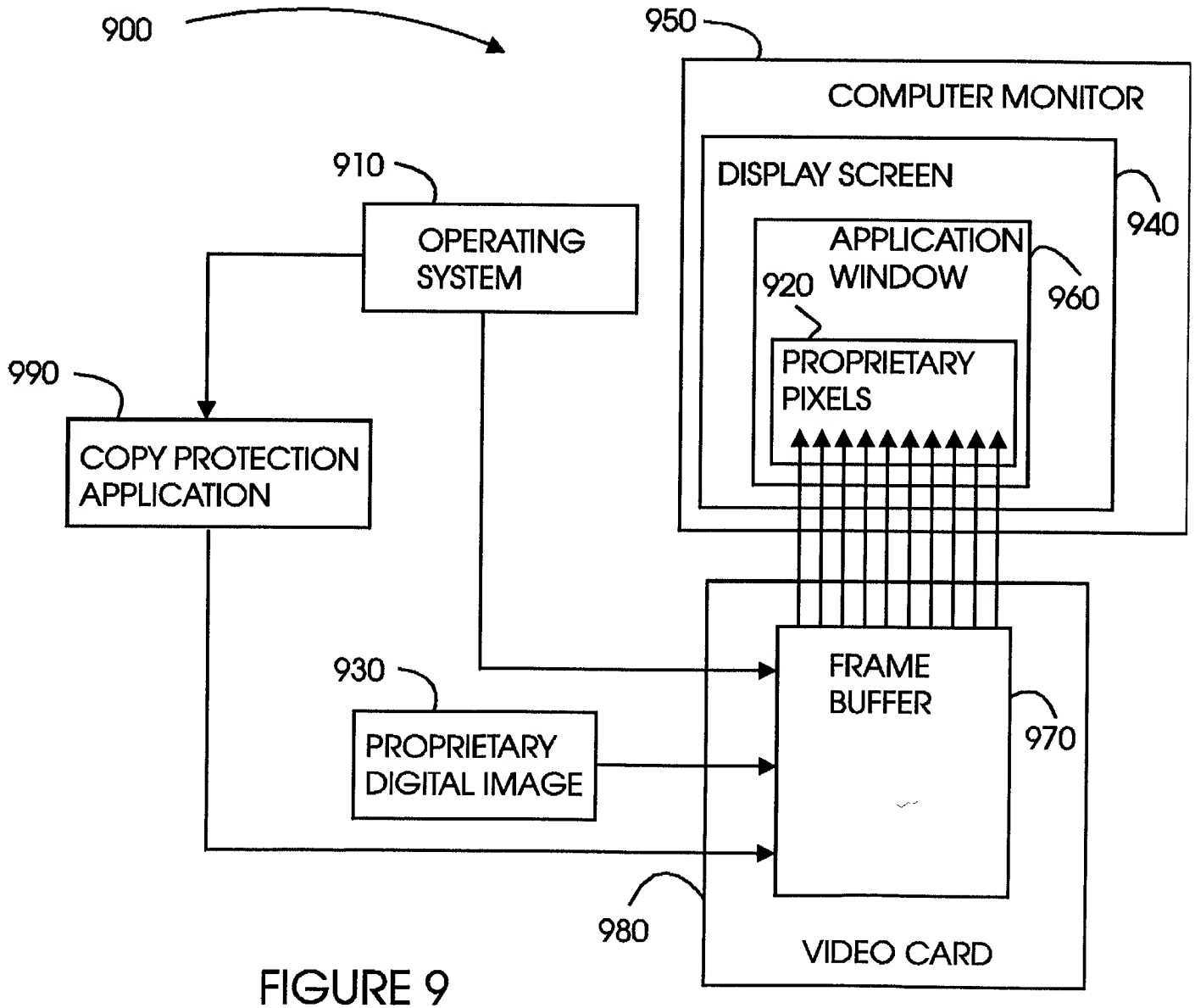


FIGURE 9

000180 "6629E960

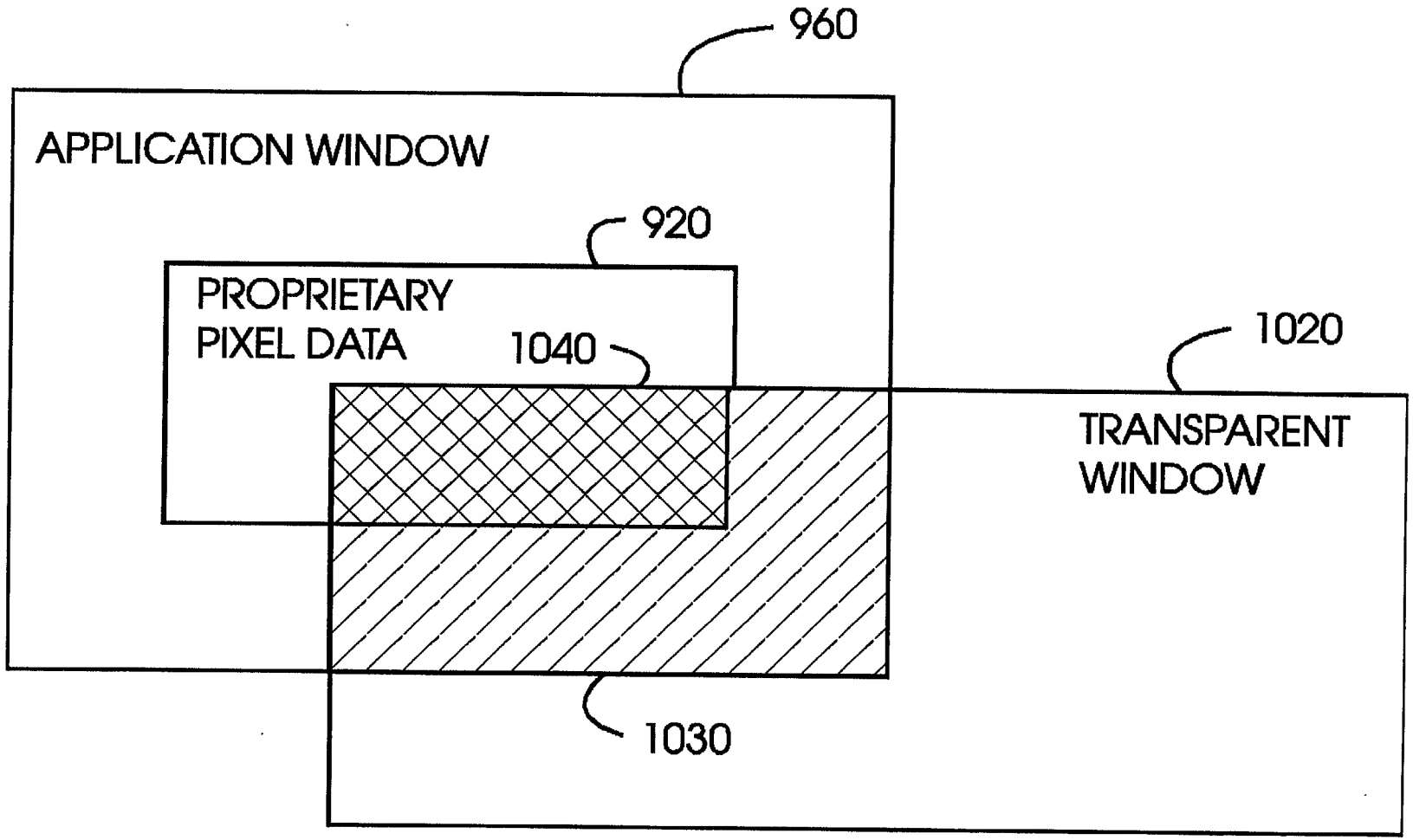


FIGURE 10

PTO/SB/01 (6–95) (modified)
Approved for use through 10/31/96 OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0010/PTO
Rev. 6/95

U.S. Department of Commerce
Patent and Trademark Office

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION

[X] Declaration Submitted with Initial Filing OR [ ] Declaration Submitted after Initial Filing

Attorney Docket Number	**5111**
First Named Inventor	**Moshe Binyamin Rubin**
COMPLETE IF KNOWN	
Application Number	**Unknown**
Filing Date	**Unknown**
Group Art Unit	**Unknown**
Examiner Name	**Unknown**

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND SYSTEM FOR COPY PROTECTION OF IMAGES DISPLAYED ON A COMPUTER MONITOR

the specification of which (*Title of the Invention*)

[x] is attached hereto

OR

[ ] was filed on (MM/DD/YYYY) [_____] as United States Application Number or PCT International Application Number [_____] and was amended on (MM/DD/YYYY) [_____] (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37 Code of Federal Regulations. § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code § 119 (a)-(d) or § 385(b) of any foreign application(s) for patent or inventor's certificate, or § 365 (a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]

[ ] Additional foreign application numbers are listed on a supplemental priority sheet attached hereto:

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	
		[ ] Additional provisional application numbers are listed on a supplemental sheet attached hereto.

DECLARATION	Page 2

I hereby claim the benefit under Title 35, United States Code § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (*if applicable*)
09/459,493	**N/A**	**December 13, 1999**	**N/A**

[ ] Additional U.S. or PCT international application numbers are listed on a supplemental priority sheet attached hereto.

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Name	Registration Number	Name	Registration Number
Laura A. Majerus **Arnold de Guzman**	**33,417** **39,955**	**Greg T. Sueoka** **Daniel R. Brownstone**	**33,800** **P-46,581**

[ ] Additional attorney(s) and/or agent(s) named on a supplemental sheet attached hereto.

Please direct all correspondence to:

Laura A. Majerus
Fenwick & West LLP
Two Palo Alto Square
Palo Alto, CA 94306
U.S.A.

Telephone	**(650) 858-7152**	Fax	**(650) 494-1417**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor: [ ] A petition has been filed for this unsigned inventor

Given Name	**Moshe**	Middle Initial	**B.**	Family Name	**Rubin**	Suffix e.g. Jr.	

Inventor's Signature	Moshe B. Rubin	Date	30 July 2000

Residence: City	**Jerusalem**	State		Country	**ISRAEL**	Citizenship	**ISRAEL/USA**

Mailing Address	**Rechov Shaulson 59**
Mailing Address	**Rechov Shaulson 59**

City	**Jerusalem**	State		Zip	**95400**	Country	**ISRAEL**

[X] Additional inventors are being named on supplemental sheet(s) attached hereto

DECLARATION	**ADDITIONAL INVENTOR(S)** **Supplemental Sheet**

Name of Additional Joint Inventor, if any:		[ ] A petition has been filed for this unsigned inventor					
Given Name	**Moishe**	Middle Initial		Family Name	**Halibard**	Suffix e.g. Jr.	
Inventor's Signature				Date	30 July 00		
Residence: City	**Jerusalem**	State		Country	**ISRAEL**	Citizenship	**ISRAEL/ British**
Mailing Address	**BenZion 2/32**						
Mailing Address	**BenZion 2/32**						
City	**Jerusalem**	State		Zip	**95423**	Country	**ISRAEL**

Name of Additional Joint Inventor, if any:		[ ] A petition has been filed for this unsigned inventor					
Given Name		Middle Initial		Family Name		Suffix e.g. Jr.	
Inventor's Signature				Date			
Residence: City		State		Country		Citizenship	
Mailing Address							
Mailing Address							
City		State		Zip		Country	

Name of Additional Joint Inventor, if any:		[ ] A petition has been filed for this unsigned inventor					
Given Name		Middle Initial		Family Name		Suffix e.g. Jr.	
Inventor's Signature				Date			
Residence: City		State		Country		Citizenship	
Mailing Address							
Mailing Address							
City		State		Zip		Country	

Name of Additional Joint Inventor, if any:		[ ] A petition has been filed for this unsigned inventor					
Given Name		Middle Initial		Family Name		Suffix e.g. Jr.	
Inventor's Signature				Date			
Residence: City		State		Country		Citizenship	
Mailing Address							
Mailing Address							
City		State		Zip		Country	
[ ] Additional inventors are being named on supplemental sheet(s) attached hereto							

09633592 .081700